

# Orion Entry, Descent, & Landing Simulation

Brian R. Hoelscher<sup>1</sup>

*NASA Johnson Space Center, Houston, Texas 77058*

The Orion Entry, Descent, and Landing simulation was created over the past two years to serve as the primary Crew Exploration Vehicle guidance, navigation, and control (GN&C) design and analysis tool at the National Aeronautics and Space Administration (NASA). The Advanced NASA Technology Architecture for Exploration Studies (ANTARES) simulation is a six degree-of-freedom tool with a unique design architecture which has a high level of flexibility. This paper describes the decision history and motivations that guided the creation of this simulation tool. The capabilities of the models within ANTARES are presented in detail. Special attention is given to features of the highly flexible GN&C architecture and the details of the implemented GN&C algorithms. ANTARES provides a foundation simulation for the Orion Project that has already been successfully used for requirements analysis, system definition analysis, and preliminary GN&C design analysis. ANTARES will find useful application in engineering analysis, mission operations, crew training, avionics-in-the-loop testing, etc. This paper focuses on the entry simulation aspect of ANTARES, which is part of a bigger simulation package supporting the entire mission profile of the Orion vehicle. The unique aspects of entry GN&C design are covered, including how the simulation is being used for Monte Carlo dispersion analysis and for support of linear stability analysis. Sample simulation output from ANTARES is presented in an appendix.

## Nomenclature

<i>ANTARES</i>	=	Advanced NASA Technology Architecture for Exploration Studies simulation
<i>API</i>	=	Application Programming Interface
<i>CAP</i>	=	CEV Aerosciences Project
<i>CEV</i>	=	Crew Exploration Vehicle
<i>CM</i>	=	Crew Module
<i>DOUG</i>	=	Dynamic Onboard Ubiquitous Graphics
<i>EDGE</i>	=	Engineering DOUG Graphics for Exploration
<i>EDL</i>	=	Entry, Descent, and Landing
<i>EI</i>	=	Entry Interface
<i>GN&amp;C</i>	=	Guidance, Navigation, and Control
<i>GPS</i>	=	Global Positioning System
<i>GRAM</i>	=	Global Reference Atmosphere Model
<i>IMU</i>	=	Inertial Measurement Unit
<i>JEOD</i>	=	JSC Engineering Orbital Dynamics
<i>JSC</i>	=	Johnson Space Center
<i>NASA</i>	=	National Aeronautics and Space Administration
<i>RCS</i>	=	Reaction Control System
<i>SM</i>	=	Service Module

## I. Introduction

When the Orion project kicked off in early 2005 with Johnson Space Center (JSC) leading the Guidance, Navigation, and Control (GN&C) design effort for the new manned space vehicle, work began almost immediately to plan and develop a new simulation tool for GN&C analysis. The project provided a rare opportunity to create a simulation with a “clean sheet” approach for a program expected to be in service for many decades. This

<sup>1</sup> CEV Entry Simulation Lead, Integrated GN&C Branch, NASA Johnson Space Center, Mail Stop - EG4.

does not happen very often in the manned space vehicle arena and much care needed to be taken in selecting a proper approach and architecture. Of course, decades of GN&C analysis supporting the Space Shuttle, Space Station, and other vehicle programs provided a rich legacy of simulation capabilities to leverage from for creating this new Orion vehicle simulation.

This paper will focus on the GN&C simulation of the Orion Entry, Descent, and Landing (EDL) flight phases, but the EDL simulation is part of a much broader effort to simulate all aspects of the Orion Project in one simulation. The EDL simulation shares its architecture and modeling software with other simulations supporting ascent, ascent abort, on-orbit, lunar orbit, and trans-lunar GN&C analysis. The simulations are constructed such that end-to-end simulation is straight-forward to setup.

The decisions made in creating the Orion EDL simulation were based on lessons learned experience with previous manned space vehicle simulations and on the anticipated needs and demands to be placed on the simulation by the user community. A key objective from the start was to ensure that only one simulation would be built due to the expense and manpower necessary to support multiple large complex simulation tools performing duplicate capabilities. The goal is to maintain one GN&C simulation at NASA to support design analysis, vehicle performance analysis, crew training, avionics-in-the-loop testing, flight software verification, etc. The expectation was that the Orion prime contractor would provide the primary independent GN&C simulation to serve as the project's check for verification and validation of simulation results.

This paper will discuss in detail the design architecture and modeling components of the NASA Orion EDL GN&C simulation tool. Sample simulation output data of a nominal skip entry trajectory will be provided in the appendix. The tool was originally named the Architecture for Exploration Studies (ARES) simulation. But when the Constellation Program announced that the Crew Launch Vehicle would be named Ares, the simulation name was changed to avoid confusion. The current name is the Advanced NASA Technology Architecture for Exploration Studies (ANTARES) simulation.

## **II. General Simulation Architecture**

### **A. Computer Lab Environment**

The computer lab environment selected to host the Orion GN&C simulations is within the JSC Aeroscience & Flight Mechanics Division. It is a facility utilizing a cluster of personal computer processors running the Linux operating system (CentOS 4.4) within a blade server for accessing mass storage. Recent experience has demonstrated that this type of lab setup provides high computational performance with excellent stability for a large group of users. The system is open architecture enabling the implementation of regular upgrades to the operating system and processor hardware and has the added benefit of being relatively inexpensive to operate and maintain.

The ClearCase<sup>1</sup> commercial-off-the-shelf (COTS) application from Rational Software is used to provide simulation software configuration management and to provide a user environment for development and analysis. ClearCase provides excellent software version tracking, a system for developers to support parallel development activities, and the means for collaborative software sharing within a large group of users.

The ClearDDTS<sup>2</sup> (distributed defect tracking system), also from Rational Software, is used to provide an issues database for all discrepancies and change requests for the simulation tool. ClearDDTS is used to record information about which software is being updated and the testing that was performed for the new software. Our computer lab is presently being upgraded to a new tracking system called ClearQuest, which will provide a tracking system that is fully integrated into the ClearCase application. ClearDDTS is currently an independent application.

This computer lab environment, configuration management tool, and discrepancy tracking tool has been used at JSC for many years with favorable results. The Orion simulation project was added to the framework with no difficulty and has been in successful operation for a couple of years. Numerous websites, Wiki forums, and document sharing resources have been created to support the simulation users and developers. It has also been successfully extended to support users and developers at other NASA centers who are supporting the Orion Project.

### **B. Simulation Backbone**

Historically, large simulation tools were created with an entire simulation architecture that was unique to the respective tool. The simulation maintained executive model schedulers, tools for input/output, graphical user interfaces, communications interfaces, data post-processing tools, math utility libraries, etc. A consequence of this "stovepiped" approach is higher cost maintenance overhead and reduced ability to port software models from one simulation to another.

Over the past two decades, the JSC Engineering Directorate has nurtured a dedicated group whose purpose is to develop and maintain a standardized simulation architecture toolkit for use in various simulations. The toolkit is

called “Trick”<sup>3</sup> and is maintained by a small group within the Automation, Robotics, and Simulation Division. The Trick toolkit has matured into a stable powerful capability that has become widely used as the simulation backbone for numerous JSC simulation applications.

By utilizing Trick, a simulation project gets a toolkit of common standardized software that serves as the backbone architecture of their simulation tool. The Trick toolkit eliminates a lot of the drudgery associated with simulation construction. It provides a framework to work within that enables the developer to get directly to the task of implementing math models. It is the “empty shell” into which math models are poured.

By the time the Orion project kicked off, JSC had migrated numerous large simulations supporting the Space Shuttle, Space Station, and other programs to the Trick architecture. A large body of Trick compliant math model software was already in existence. It was a natural decision to select Trick for the ANTARES simulation so the project could leverage off of the existing model base.

The math model software base has been segregated utilizing ClearCase into various software packages. For example, there is a defined package for the Trick software and a separate package for the generalized dynamics and environment models. A “common model library” package was created to store software of a generalized nature that could find application in numerous simulation applications. Finally, unique software packages are defined for software specific to a particular simulation or vehicle design such as the Orion vehicle.

### **C. Dynamics Package**

The modeling of six degree-of-freedom dynamics of aerospace vehicles is fairly universal and independent of the configuration of the vehicle. Generalized dynamics formulations and earth environment modeling have been used with great success in a number of simulation applications. As the Trick toolkit became widely used, it was natural that a Trick-compliant generalized dynamics package would evolve and also become widely used in the various Trick-based simulators. This dynamics package is named the JSC Engineering Orbital Dynamics (JEOD) package<sup>4</sup> Version 1.5. JEOD is officially maintained by the JSC Automation, Robotics, & Simulation Division. The ANTARES team concluded that the JEOD package was sufficiently mature and capable to be adopted for the Orion simulation.

The JEOD package provides several useful capabilities for the Orion EDL simulation that are summarized here:

#### *1) Generalized multi-body dynamics formulation*

The core dynamics of JEOD is a multi-body six degree-of-freedom dynamics based on generalized inertial coordinates. The dynamics can propagate N independent free-flying bodies using a rigid body dynamics formulation. The user has the option to select the type and fidelity of the integrator utilized for a given problem. The bodies can be attached and detached during runtime with the new states after an attachment/detachment event generated based on conservation of momentum principles. For combined bodies, the inertial state of the composite center of mass is propagated by the integrator, while the inertial state of each constituent body is always calculated based on kinematics. So, a continuous state of the constituent bodies is always available throughout a series of attach/detach events.

For the Orion EDL application, most analysis is performed using a single vehicle formulation of the CEV Crew Module (CM) trajectory from Entry Interface (defined as an altitude of 400,000 ft) to its landing on the ground. However, some analysis has been setup with the simulation as a multi-body formulation to model CEV mass properties changes as components are jettisoned during the entry trajectory and to support footprint analysis of the components being jettisoned. Examples of jettisoned components are the Service Module, Forward Blast Cover, Drogue Chutes, and the Heatshield.

#### *2) Generalized mass properties calculation*

The JEOD package includes a mass tree model where a composite body can be represented as an attachment of N mass bodies. The topology is constructed using a linked list approach where one body is designated as the parent body and child bodies are attached to the parent. The link list is traversed to calculate the composite mass properties of the parent body for use in the dynamics formulation. The mass properties of any component mass body can be dynamically updated with continuous updates to the composite mass properties automatically calculated. Any pair of bodies in the list can be attached or detached at any time with new composite mass properties calculated after each attach/detach event.

For the Orion EDL application, dynamic mass depletion would be modeled to account for Reaction Control System (RCS) fuel depletion and heat ablation of the thermal protection system. The capability to detach mass bodies from the parent CEV CM body would be utilized to model the jettison of the Service Module, Forward Blast Cover, Drogue Chutes, and the Heatshield.

### 3) *Earth and planetary environment*

The JEOD package provides earth and planetary environment models to model gravitational fields, ephemeris data, universal time, and planetary rotation, nutation, and precession (RNP). The model also includes the ability to transition from the gravitational field and inertial reference system of one planetary body to another (ie. transition from earth gravity to lunar gravity for a trans-lunar orbit).

For the Orion EDL application, we have configured the environment to utilize the gravitational effects of three planetary bodies – Earth, Moon, and Sun. The Earth gravitational field is the NASA Goddard Spaceflight Center and National Imagery and Mapping Agency Joint Geopotential Model (EGM96)<sup>6</sup> configured with 8X8 coefficient data. The Sun and Lunar third body effects are simple spherical gravity models. The Earth RNP model is for the Standard Epoch – J2000<sup>7</sup>. Data from the NASA Jet Propulsion Laboratory Planetary and Lunar Ephemerides Model (DE405)<sup>8</sup> are utilized.

## **D. Visual Graphics Package**

Another commonly used software package within Trick simulations selected for ANTARES is the Engineering DOUG Graphics for Exploration (EDGE) version 1.0<sup>5</sup>. EDGE provides ANTARES users visualization graphics of the Orion vehicles during their flight trajectory. EDGE is an adaptation of the Dynamic Onboard Ubiquitous Graphics (DOUG) used by the Space Shuttle / Space Station programs. EDGE is maintained by the JSC Automation, Robotics, & Simulation Division.

## **III. Orion Entry, Descent, & Landing Simulation Components**

This section covers a number of topics and models that are somewhat unique to the EDL simulation problem. It includes initialization options, state definitions, aerodynamics, atmospheric models, heating models, and hardware models such as parachutes, sensors, and effectors. EDL GN&C is a separate major topic that will be discussed in section IV.

### **A. Planet Relative States, Atmosphere Relative States, and State Initialization Options**

Since the JEOD dynamics package is a generalized inertial dynamics model, some kinematic dynamics models are needed to support the entry problem. The vehicle state is calculated with respect to the planet fixed frame. From this is derived the altitude, altitude rate, latitude, and longitude in the geodetic and geocentric frames. We also calculate entry trajectory information (velocity, flight path angle, and azimuth) in the topocentric and topodetic frames for both an inertial reference and a planet fixed reference. For nominal entry trajectories, we know the target landing point, so actual down range and cross range distances are calculated.

States are also needed with respect to the free stream flow for modeling of aerodynamic effects. Here we calculate the free stream velocity, which takes into account the atmospheric wind vector, and we calculate the vehicle Mach number and dynamic pressure. The free stream flight path angle and azimuth trajectory is also determined. The orientation of the vehicle with respect to the free stream flow is determined in the form of the Euler angles – bank angle, angle of attack, and angle of sideslip. Total angle of attack is calculated. Sensed acceleration is resolved into lift, drag, and side acceleration components.

A front end was created for the JEOD dynamics state initialization since JEOD primarily supports only orbital dynamics constructs such as orbital elements, local-vertical-local-horizontal frames, and states relative to a target vehicle for rendezvous and docking scenarios. The state must ultimately be transformed into the inertial frame for the dynamics. For the entry problem, the initial state is generally specified in terms of planet relative and atmosphere relative states which need to be transformed to inertial frame. We have provided several options for state specification (including inertial frame), but the most commonly used data set is geodetic altitude, latitude, and longitude for position, and angle-of-attack, angle-of-sideslip, and bank angle for orientation.

### **B. Vehicle Configuration Manager**

A unique vehicle configuration manager was developed primarily for the ANTARES ascent simulation that has also been adapted for use in the ANTARES entry simulation. Its purpose is to manage the configuration of the aerodynamics models, the atmosphere models, and the execution of body detachment events. The vehicle configuration manager monitors GN&C state and status flags to determine the appropriate time to perform a body detachment in the JEOD mass tree model. When a body is detached, the atmosphere and aerodynamic models are activated and configured for the newly free-flying vehicle and the manager decides whether a new aerodynamics model needs to be loaded for the main CEV vehicle after an attached component has been jettisoned.

For example in an ascent scenario, the CEV is attached to the booster stages when on the launch pad. As the vehicle ascends toward orbit, first the stage 1 booster and later the stage 2 booster are separated from the vehicle. Upon each booster separation, a substantially changed aerodynamic model is needed to represent the new ascending vehicle stack. The vehicle configuration manager provides the mechanism for loading a new set of aero data and executing the separations.

For the EDL application, we have thus far represented the CEV CM with a single capsule aerodynamics model for the entire entry trajectory. We have not yet needed an ability to change the aerodynamics model during a scenario. Though we envision a possibility of needing this capability in the future to handle the changes in aerodynamics as the thermal protection system ablates and changes the mold line of the heatshield. However, we do sometimes utilize a multi-body formulation to simulate component jettisons and the vehicle configuration manager has been useful for this purpose.

Another practical consideration is the computational intensity of the atmosphere model in use in ANTARES (to be discussed in section D). Since the atmosphere outputs are only needed for the vehicles with an active aerodynamics model, we have incentive to turn off the atmosphere for vehicles that do not need it. When several bodies are attached together, only the parent body has an active aerodynamics and atmosphere model. The child bodies have their aerodynamics and atmosphere models activated only upon detachment as a free flyer. The vehicle configuration manager provides the mechanism for activating these models upon detachment.

### **C. CEV Entry Aerodynamics**

The Orion CEV aerodynamics model is being derived and developed by the CEV Aerosciences Project (CAP) within the JSC Applied Aeroscience & CFD Branch. This group has responsibility for providing aerodynamics models for the Orion Project to both government and prime contractor users. The CEV aerodynamics model is delivered in the form of a software package called the Application Programming Interface (API)<sup>9</sup>, which includes the data tables for the model. It has been incorporated into ANTARES within a simple wrapper with inputs from the atmosphere model and the atmosphere relative state model and with outputs of aerodynamic forces and moments to be applied to the vehicle dynamics.

The CEV CM aerodynamics model was initially based on Apollo aerodynamics data because, by design, the CEV capsule outer mold line shape is identical to the Apollo capsule shape. The data was initially scaled up to the size of the CEV capsule. The current aerodynamics data (version 0.25) for CEV has been further refined by computational fluid dynamics modeling. Much refinement is still to come via wind tunnel testing and test flight data analysis.

The CAP team defines the aerodynamics characteristics by specifying coordinate frames, reference length and area, the moment reference center, and tables of aerodynamics coefficients as a function of Mach number and total angle of attack. They also specify the uncertainty levels on the provided data and provide instructions on whether the data should be dispersed uniformly or by Gaussian distribution. The aerodynamic coefficients include the aero moment damping derivatives. For the CEV mold line shape, the damping derivatives can be unstable at low Mach numbers for certain angles of attack. This means CEV is not passively stable when flying heatshield forward for all stages of an entry trajectory.

The current CEV aerodynamics model neglects interaction effects due to RCS jet plume flow. As mentioned earlier, it also neglects the aerodynamic changes due to heatshield ablation. And it neglects aerodynamic separation effects while components are in proximity after jettison. Models of these effects will be developed in the future.

### **D. Earth Atmosphere**

The Orion Project has specified that the Earth atmosphere be represented by the Global Reference Atmosphere Model (GRAM) in project analysis tools. ANTARES has an implementation of the GRAM-99 Version 3 model as provided by Marshall Spaceflight Center<sup>10</sup>. GRAM-99 is a large computationally intensive Fortran language software model of the Earth's atmosphere that has traditionally not been practical to embed inside of simulation tools, especially those intended for realtime applications. Instead GRAM has been used in the past to output dispersed atmosphere table-lookup files for reference trajectories. However, we feel that computers have advanced to the point where it is now practical to incorporate the GRAM model inside GN&C analysis simulation tools.

There have been many lessons to be learned during our work to incorporate this complex Fortran model within a predominantly C language simulation environment on a Linux computer platform. The simulation has the added complexity of needing GRAM to be simultaneously active for multiple vehicles. We have been successful in our implementation with good verification results to date. Despite the computational intensity of the model, we are still able to achieve execution performance near 100 times realtime when calling GRAM at a 25 hz cycle rate.

Our simulation users have been able to take full advantage of the features of GRAM-99 in their analysis. GRAM estimates the mean values and standard deviations for atmospheric parameters such as density, temperature, and winds, for any month, at any altitude and location within the Earth's atmosphere. It provides statistically realistic variations for these parameters along any flight trajectory for both long term seasonal or regional variations in the atmosphere and short term perturbations such as wind gusts and shear. This perturbation feature makes GRAM especially useful for Monte Carlo dispersion analysis. The model also supports a range reference atmosphere feature to increase accuracy at particular launch or landing site locations.

ANTARES has other atmosphere model options available. Users can switch to the U.S. Standard 1976 atmosphere model, if desired. Users can also provide table lookup files to represent other atmosphere representations. For example, table lookup data could be used to load in balloon data for a simulation run. We have also recently incorporated a beta version of GRAM 2007 which has some unique new features that will be explored in future analysis.

#### **E. CEV Aerothermal Heating**

A model to estimate the aerothermal heat rate and integrated heat load experienced by the thermal protection system has been incorporated into ANTARES. This model was developed at NASA Langley Research Center. The CEV Entry Heating Model, version 3.2<sup>11</sup>, calculates the radiative heating rate, convective heating rate, and wall temperatures at the flow stagnation point and the hot corner point. These heat rates are integrated to estimate the total heat load at those points. The model also calculates the maximum aerothermal shear and pressure experienced on the CM vehicle. The model includes the heating margins defined by the CAP team.

#### **F. CEV Parachutes**

The ANTARES entry simulation has a simplified parachute model. It considers the chutes to be an effector model that is applying external forces and torques onto the CEV CM vehicle. The two drogue chutes and the three main chutes are modeled as a single chute of equivalent size. The mass transport during chute deployment is ignored. However, the various stages of chute deployment including reefing and inflation are modeled with a fairly high degree of accuracy. The result is a force model that is quite accurate with the translational dynamics following a trajectory that matches very well with much higher fidelity chute simulations. However, the torque models are of more limited accuracy because of the preceding approximations.

The CEV chute model used in ANTARES is derived from a chute model originally developed for the Space Shuttle drag chute deployed during landing runway rollout. The model was adapted to represent a CEV chute with appropriate CEV chute data. An early version of the model produced very poor rotational dynamics and was unstable when using the CEV Aero API version 0.25. This instability was caused by the unstable aero moment damping derivatives when at low Mach numbers. The damping terms of the chute model were subsequently improved using chute equations developed during the Apollo Project. The revised damping terms improved the rotational dynamics performance while using the Aero API to the point of being a generally good match to an independent high fidelity chute simulation.

The ANTARES simulation deploys the drogue chutes at about 40,000 ft above mean sea level (exact altitude is a function of guidance mode and targeting accuracy). This happens three seconds after the jettison of the forward blast cover. Two drogue chutes are deployed with two stages of reefing. At 8,000 ft above ground level, the drogues are jettisoned and the three main chutes are deployed with three stages of reefing. For water landing, the heatshield is retained until splashdown. For land landing, the heatshield is jettisoned 30 seconds after initial deployment of the main chutes.

ANTARES terminates execution when the CEV reaches the altitude of the landing target location. A ground impact force model has not yet been implemented. The firing of retro rockets just before landing has not been simulated (it is still not certain that CEV will have retro rockets).

#### **G. CEV Entry Sensors**

During the entry phase of the mission, the CEV CM has two primary navigation sensors – an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS) signal receiver. Both sensors are modeled utilizing a “generic” sensor construct with shared software. This generic sensor approach has been adapted for use in a large number of sensor models in ANTARES across the ascent, on-orbit, entry, and lunar orbit simulations. The IMU and GPS measurement outputs are provided as inputs to the CEV navigation algorithms.

The IMU model simulates the errors generated by the signals from the gyros and accelerometers of a strapdown IMU system. It includes the effects of misalignment, bias, scale factor errors, white noise, random walk, and quantization. The IMU model integrates the actual body angular rate and actual inertial translational acceleration to

output an accumulated delta angular rotation and delta translational velocity that has been subjected to the error effects.

Similarly, the GPS model simulates the estimated noise and bias associated with a position and velocity measurement from a GPS receiver. It outputs a measurement of the inertial position and velocity of the vehicle calculated from the actual state based on the error effects. GPS blackout periods are simulated in ANTARES and are triggered based on altitude and drag. GPS signals are lost when aerodynamic drag exceeds about 0.05 “g”. GPS signals are restored when the altitude is either below 180,000 ft or the drag is again below 0.05 “g” in case of a skip trajectory. ANTARES can model cases where GPS fails to reacquire a signal after a blackout. We are actively looking at deriving more realistic methods of modeling the blackout periods, possibly looking at aerothermal heat rates as the trigger criteria.

Both the IMU and GPS sensor models will likely be upgraded once vendor selections are made and high-fidelity specific models can be defined. The high-fidelity models may come from the hardware vendor. We currently only model one IMU unit, but the actual vehicle will have four units plus one for emergency mode. At some point, we will model all of the IMU units to test fault tolerance. Also, we would like to implement a GPS satellite constellation model to better simulate the true GPS signals and their availability to the GPS receiver.

The CEV CM is likely to have additional sensors as the GN&C design matures. A flush air data system is under consideration to provide an altitude measurement based on barometric pressure. This would be a backup in case the GPS receiver fails to pick up a signal after blackout. Radar altimeters may also be needed to ensure the chutes and airbags deploy at safe altitudes to protect against the possibility of large navigated altitude errors that might result if GPS signals are missing. These sensor models will be developed as needed for analysis.

## **H. CEV Control Effectors**

The CEV CM has one control effector – the Reaction Control System (RCS) jet thrusters. ANTARES utilizes a “generic” RCS model that was originally adapted from a Space Shuttle RCS simulation model and has been employed to simulate the RCS model of several space vehicles in recent years. In ANTARES, the same generic RCS model is used for both the Crew Module RCS and the Service Module RCS. The model receives jet commands from flight control and outputs a force and torque that is applied to the vehicle dynamics.

The generic RCS model has many features. It models N thrusters with thrust build-up and trail-off, signal delays between command and actuation, thrust reduction based on fuel rates, self impingement, alignment errors, blow down effects, minimum jet on-times and off-times, and fuel mass consumption. At the current level of CEV RCS design maturity; we are not yet utilizing all of these capabilities. We have performed studies evaluating signal delays and jet minimum on/off times. We have looked closely at the fuel consumption output during our analysis studies. But we are not yet depleting the consumed mass from the mass properties of the CEV vehicle. Mass depletion is a capability to be added soon.

The CM has three strings of six 160 pound methane/oxygen thrusters (vacuum thrust). The three strings provide tolerance to two RCS faults. Our analysis requires that the system have adequate control authority with one string of jets for all trajectories and expected dispersions. One string provides one jet for each attitude rotation direction (+/- roll, +/- pitch, and +/- yaw). Our default analysis configuration sets the specific impulse equal to 305.6 seconds, the signal delay to 80 milliseconds, the minimum jet on-time equal to 60 milliseconds, and the minimum jet off-time equal to 25 milliseconds. The model is configured with an additional model to simulate the effect that static atmospheric pressure has on jet thrust. The thrust output is reduced from the vacuum rating when the jet is fired in the atmosphere.

We have also configured the RCS model to simulate firing the thrusters in a “cold gas” mode, which is a dissimilar use of the thrusters in case of a fault with the thruster igniters. There are several modes of cold gas operation, all of which produce thrust that is less than 40 pounds with a specific impulse of less than 100 seconds. We have demonstrated fairly good success at controlling the vehicle in emergency ballistic entry mode at this low thrust level.

Similar to the sensor modeling, the RCS model will be upgraded as detailed hardware knowledge is provided by the RCS vendor. The goal will be to implement an RCS model of the highest possible fidelity to match actual flight performance.

## **IV. Guidance, Navigation, & Control for Orion Entry, Descent, & Landing**

The primary purpose of the ANTARES simulation is to support GN&C design and analysis with a six degree-of-freedom simulation tool. This section discusses the GN&C capabilities of ANTARES as pertains to the Orion EDL

application. Before getting into the discussion of the GN&C algorithms, we will first discuss the unique architecture which enables maximum flexibility in testing various GN&C algorithms.

## **A. GN&C Architecture**

When the development of ANTARES began at the start of the Orion Project, the initial GN&C prototype implementations were simple. Simple generic controllers were implemented with perfect navigation, perfect sensors, and perfect effectors. The initial entry guidance was an implementation of the original Apollo guidance algorithm. The team knew that over the course of the Orion Project, we would implement many versions of the GN&C component algorithms for design and testing evaluation. Ultimately, a point would be reached where the simulation would utilize actual CEV GN&C flight software. It was important to retain the ability to test flight software against the original engineering design algorithms. We sought maximum flexibility in our GN&C architecture to support these simulation requirements. We have been successful in achieving this goal.

The key components of the ANTARES GN&C architecture are: 1) a GN&C executive with generalized state and status information; 2) a set of manager functions for each component of navigation, guidance, and control; 3) detailed mode definitions for GN&C algorithms applicable to various flight phases; 4) generalized output data sets for each component (sensors, navigation, guidance, control, effectors); and 5) a simulation configuration selector to provide the user a mechanism for configuring the simulation to a particular setup<sup>12</sup>.

### *1) GN&C Executive*

The GN&C executive is geared toward supporting the entire CEV GN&C capability for ascent, on-orbit, entry, and lunar phases of flight. The entry capabilities within the executive are a small subset of the overall executive implementation. The same executive function is shared by the ANTARES simulations used to support analysis in each of these functional areas. This commonality will ease the migration to an end-to-end mission GN&C simulation when the need arises.

The GN&C executive is fundamentally a task manager. At a given point in the CEV mission, a set of tasks is loaded into the executive via task data. The executive evaluates the task completion criteria for each task to determine when the task is completed. Once completed, the executive advances to the next task in the list. When the last task in the list is reached, new task data is loaded to provide information regarding the next list of tasks. The executive has numerous options for overriding the task information (ie. crew input, ground input, abort decision logic, etc.). Each task provides a mechanism to load new GN&C algorithm data to customize GN&C performance during the task execution.

GN&C executive tasks are grouped according to a mission hierarchy. At the highest level is the set of CEV mission phases. Phases include Earth Ascent, Earth Orbit, Earth Entry, Lunar Orbit, Earth-Lunar Trans-Orbit, etc. All Orion EDL simulations exclusively utilize the Earth Entry phase. The next level of task hierarchy is the mission segment. Each phase can have numerous mission segment options, some of which are for nominal mission operations while others are for abort mission operations. The Orion EDL simulation presently sequences through three nominal GN&C segments – Nominal Entry, Chute Sequence, and Landing. Each segment has unique GN&C capability requirements. For example, during the chute sequence segment the CEV is unguided and uncontrolled (it is only navigating). During the landing segment, the CEV is still unguided, but a specialized controller is utilized to provide roll control under the main chutes to orient the vehicle along the wind drift direction. The chute sequence and landing segments are common to nominal and abort entry as well as ascent abort scenarios. In addition to the nominal GN&C segments, two entry abort segments are supported. The first is a ballistic unguided entry configuration where the vehicle spins about the bank axis during the entire entry trajectory. The ballistic abort mode is the last-ditch emergency mode of return to the Earth's surface. The second is a constant bank unguided entry configuration where the vehicle flies a user-specified constant bank angle or canned bank angle profile. The constant bank abort mode is commonly used following ascent aborts.

Finally, the lowest level of mission hierarchy is the task definition. Each segment can support a list of N tasks. Each task defines a GN&C mode to fulfill its purpose. For each mode, sub-modes are defined for navigation, guidance and control to determine which particular algorithm will be executed for the specified task. Each task loads in a new set of data to configure the specified GN&C algorithms. The task data typically specifies the task completion criteria for the executive. Several forms of task completion criteria can be defined. A task could complete after a specified time duration, or be triggered by a navigated state value such as altitude, or be triggered by a status flag output by GN&C.

### *2) GN&C Manager Functions*



The GN&C algorithms are executed within a series of manager functions. There is a manager function for navigation algorithms, guidance algorithms, and control algorithms. Manager functions are segregated by execution frequency (rate groups). For example, the control algorithm might be separated into a high rate inner loop controller and a medium rate outer loop controller. Guidance could be setup to run at a medium or slow rate of execution. Also, multiple managers supporting independent algorithms can be implemented into ANTARES with options for activating a particular version for a given scenario. For example, a perfect navigation manager can co-exist with a high-fidelity engineering navigation manager and with an actual flight software navigation manager. The user chooses which navigation manager to activate.

The main purpose of a GN&C manager is to select and execute a particular GN&C algorithm. Each manager will contain a full complement of GN&C functions for ascent, on-orbit, entry, etc. The function executed is determined by the task data that configures a particular navigation, guidance, or control sub-mode for a specified overall GN&C mode. The GN&C managers are also responsible for setting the state and status data needed by the executive to regulate task flow.

### *3) GN&C Mode Definitions*

The GN&C modes define every unique period of the Orion vehicle concept of operations. For each mode, a set of sub-modes is defined to configure the particular algorithms that will be used for navigation, guidance, and control. Each GN&C mode does not represent a unique algorithm set. Modes can be defined to represent a change in GN&C configuration data based on a trigger. For example, the entry attitude control algorithm may be identical for exo-atmospheric flight and atmospheric flight, but the control gains may be tuned differently for these different flight regimes. Two different modes may be using the same GN&C algorithm set, but the configuration data is different for the two flight regimes. Another situation arises where a mode is defined to trigger an event such as main chute deployment. The GN&C algorithm set and configuration data may be identical before and after main chute deployment, but a separate mode is defined to support the signals needed by the executive task management to command the deployment of the main chute.

### *4) Generalized Output Data*

A generalized set of output data structures has been created to support the ANTARES GN&C architecture. Output structures have been defined for each type of sensor, navigation, guidance, and control. Each output structure contains a set of data that has been generalized to be applicable to similar algorithm types. As new GN&C algorithms are implemented into ANTARES, the outputs are either generated internally within the generalized output structure or the data is loaded into the generalized output structure by the GN&C manager functions. In this way, the generalized sensor output structures (GPS and IMU) are passed into all navigation managers, the generalized navigation output structure is passed into all guidance and control managers, and data from the generalized control output structure is passed to the effector models (RCS). No GN&C manager functions have a hard software dependency on the data structures of a particular external algorithm (but do depend on the structures of the internally called algorithms). This approach provides flexibility in implementing multiple algorithms of varying levels of fidelity with minimal software cross-dependency. All algorithms write to the same output data structure and data flows cleanly through the system from sensor to navigation to guidance to control to effector.

### *5) GN&C Simulation Configuration Selector*

The simulation configuration selector is the real magic that ties the whole GN&C architecture together and makes this approach powerful for the GN&C designer or analyst. The architecture permits multiple versions of every capability to co-exist within the simulation tool. Each capability writes its output to a common generalized data structure, which is then passed as an input to the next piece in the GN&C system logic flow.

The user has two tools to select which algorithms will execute and which data will flow through the models. First the user has activation flags to turn on or off the various versions of each capability. For example, three navigation managers might exist within the simulation – a perfect navigation model, a high-fidelity engineering navigation model, and an actual flight software navigation function. The user would typically activate one of the three navigation managers and turn the others off. Or, the user might choose to run two navigation algorithms simultaneously. One algorithm would have its data configured to close-the-loop with the rest of GN&C, while the other algorithm would process the same inputs but would execute in “shadow” mode. The algorithm running in shadow mode would have no effect on the closed-loop performance of the GN&C system, but its output could be compared to the algorithm that had been used to close-the-loop. In this way the user can verify the performance of flight software compared to the original engineering algorithm.

The second tool is a set of structure pointers to the generalized output structures. Since multiple versions of every sensor, effector, or GN&C manager may exist in the simulation, each version utilizes a structure pointer to the generalized output structure as a calling argument. The user must configure in their simulation input file which actual structure will be assigned to the simulation configuration pointers. In this way, the user defines which model's output structure will close-the-loop with the rest of the GN&C system.

The result is a very flexible architecture that enables the testing of multiple algorithm versions within one simulation tool with the capability of running more than one algorithm at the same time to assess relative performance.

## B. CEV Entry GN&C Executive Logic and Task List Definition

The preceding section described the general GN&C architecture of the ANTARES simulation. This section will lay out some detail about how the architecture has been configured for the Orion EDL simulation. The following lists describe all of the GN&C phases, segments, and modes utilized by the EDL simulation. They provide some information about the task lists defined for each segment and also give an idea of what sensors, GN&C algorithms, and effectors are in use, including information about the completion criteria for each task. The Orion EDL simulation is a rapidly evolving simulation and this information is a snapshot in time.

### 1) *Phases*

cev\_PHS\_ENTRY\_EARTH – The Orion EDL simulation only deals with earth entry scenarios.

### 2) *Segments*

cev\_SEG\_ENTRY\_NOM – Nominal entry from EI to chute deployment

cev\_SEG\_CHUTE\_SEQUENCE – Nominal drogue and main chute deployment periods

cev\_SEG\_LANDING – Nominal landing attenuation and vehicle control

cev\_ABT\_ENTRY\_BALLISTIC – Abort unguided entry with controlled ballistic bank angle spin rate

cev\_ABT\_ENTRY\_MAX\_LIFT – Abort unguided entry with controlled constant bank angle

### 3) *GN&C Modes*

cev\_EXOATMOS\_ENTRY

- Task #1 of nominal entry segment
- Starts at entry interface (400,000 ft altitude); ends when 0.05 'g' of drag is detected
- IMU & GPS sensors, hifi navigation, entry guidance, 3-axis control, RCS effector

cev\_SKIP\_GUIDED\_ENTRY\_BLACKOUT

- Task #2 of nominal entry segment
- Starts at onset of drag; ends when drag is below 0.05 'g'
- Note: If guidance is not performing a skip trajectory, executive jumps to task #4.
- IMU sensor, hifi navigation, entry guidance, roll control with pitch/yaw damping, RCS effector

cev\_SKIP\_EXOATMOS\_ENTRY

- Task #3 of nominal entry segment
- Starts when drag is below 0.05 'g'; ends when drag is again above 0.05 'g'
- IMU & GPS sensors, hifi navigation, entry guidance, 3-axis control, RCS effector

cev\_GUIDED\_ENTRY\_BLACKOUT

- Task #4 of nominal entry segment
- Starts at onset of drag; ends when drag altitude is below 180,000 ft
- IMU sensor, hifi navigation, entry guidance, roll control with pitch/yaw damping, RCS effector

cev\_GUIDED\_ENTRY\_WITH\_GPS\_NAV

- Task #5 of nominal entry segment
- Starts at 180,000 ft; ends when velocity is subsonic
- IMU & GPS sensors, hifi navigation, entry guidance, roll control with pitch/yaw damping, RCS

cev\_GUIDED\_ENTRY\_SUBSONIC

- Task #6 of nominal entry segment
- Starts at when velocity is subsonic; ends when guidance commands chute deployment
- IMU & GPS sensors, hifi navigation, entry guidance, roll control with pitch/yaw damping, RCS
- Note: Control gains are changed to better handle aerodynamic instabilities when at low speed.

cev\_START\_BALLISTIC\_ENTRY

- Task #1 of ballistic entry abort segment
  - Starts at abort initiation; ends when vehicle is reoriented to entry attitude
  - IMU sensor, simple navigation, unguided, 3-axis control, RCS effector
- cev\_EXOATMOS\_BALLISTIC\_ENTRY
- Task #2 of ballistic entry abort segment
  - Starts after vehicle reorientation; ends when drag is above 0.05 'g'
  - IMU sensor, simple navigation, unguided, 3-axis bank rate control, RCS effector
- cev\_UNGUIDED\_BALLISTIC\_ENTRY
- Task #3 of ballistic entry abort segment
  - Starts when drag is above 0.05 'g'; ends at chute deploy altitude of 45,000 ft
  - IMU sensor, simple navigation, unguided, bank rate control with pitch/yaw damping, RCS
- cev\_EXOATMOS\_MAX\_LIFT\_ENTRY
- Task #1 of constant bank entry abort segment
  - Starts at abort initiation; ends when drag is above 0.05 'g'
  - IMU sensor, simple navigation, unguided, 3-axis control, RCS effector
- cev\_UNGUIDED\_MAX\_LIFT\_ENTRY
- Task #2 of constant bank entry abort segment
  - Starts when drag is above 0.05 'g'; ends at chute deploy altitude of 45,000 ft
  - IMU sensor, simple navigation, unguided, bank control with pitch/yaw damping, RCS effector
- cev\_CHUTES\_FBC\_JETTISON
- Task #1 of nominal chute sequence segment
  - Starts at chute sequence initiation; ends three seconds later
  - Commands the jettison of the forward blast cover
  - IMU & GPS sensors, hifi navigation, unguided, uncontrolled
- cev\_CHUTES\_DROGUE\_DEPLOY
- Task #2 of nominal chute sequence segment
  - Starts three seconds after chute sequence initiation; ends at 8000 ft above ground level
  - Commands the deployment of the drogue chutes
  - IMU & GPS sensors, hifi navigation, unguided, uncontrolled
- cev\_CHUTES\_MAIN\_DEPLOY
- Task #3 of nominal chute sequence segment
  - Starts at 8000 ft above ground level; ends 30.2 seconds later
  - Commands the jettison of the drogue chutes and the deployment of the main chutes
  - IMU & GPS sensors, hifi navigation, unguided, uncontrolled
- cev\_CHUTES\_HS\_JETTISON
- Task #4 of nominal chute sequence segment
  - Starts 30.2 seconds after main chute deployment; ends at 1500 ft above ground level
  - Commands the jettison of the heatshield
  - IMU & GPS sensors, hifi navigation, unguided, uncontrolled
- cev\_CHUTES\_ACTIVE
- Task #1 of nominal landing segment
  - Starts at 1500 ft above ground level; ends at landing on ground (or water)
  - IMU & GPS sensors, hifi navigation, unguided, roll control under main chute, RCS effector

### C. CEV Entry Navigation

Four entry navigation algorithms have been implemented into ANTARES. The first is a perfect navigation model which simply transfers the actual states from the dynamics models to the generalized navigation output structure for use in guidance and control. The second is a pseudo navigation model. Pseudo navigation sets up a “generic sensor” to add noise and bias to the actual dynamics states so that the navigated state is a somewhat realistic representation of the expected navigation performance of the overall system. The third is the X-38 vehicle entry navigation algorithm. This software was originally based on Space Shuttle entry navigation but was adapted for the X-38 test vehicle by engineers at Johnson Space Center. It was adapted again as an early prototype of CEV

entry navigation. The fourth is the current development version of the CEV high-fidelity entry navigation algorithm designed at JSC for navigation performance analysis.

The current entry navigation algorithm uses Kalman filtering techniques to propagate the vehicle states for inertial position, inertial velocity, and the attitude error quaternion. The attitude error quaternion is propagated from IMU gyro rate measurements. The inertial position and velocity is propagated from IMU accelerometer measurements. The propagator uses a Runge-Kutta 4 integration method and utilizes a 2x2 gravity model. The navigated states for position and velocity are updated by GPS measurements. A separate navigation filter is being developed to use accelerometer data to estimate aerodynamic and atmospheric parameters for use within the entry guidance algorithms. A flush air data system atmospheric pressure transducer may be needed to stabilize the altitude estimate for situations where GPS signals might not be reacquired after a blackout period.

ANTARES has an initialization feature that allows the user to input the expected navigation filter covariance matrix at entry interface. This covariance matrix can be used to initialize correlated dispersions for both the initial navigated state and the actual state within the dynamics model. This is the most realistic approach to modeling initialization dispersions. The alternative is to directly apply random uncorrelated dispersions to the initial state variables. However, the challenge is in performing the analysis necessary to define an expected covariance matrix. ANTARES has been used to simulate navigation performance for lunar return trajectories starting 40 minutes prior to entry interface with the objective of defining the expected navigation errors at entry interface. This knowledge is used to create better initialization dispersions for atmospheric entry analysis.

#### **D. CEV Entry Guidance**

Several entry guidance algorithms have been implemented into ANTARES. The initial entry guidance implementation was a version of the original Apollo entry guidance algorithm. The Apollo guidance was replaced by two JSC designed entry guidance algorithms. One is a modified Apollo guidance adapted for the CEV vehicle size that is used primarily for scenarios involving a return from low Earth orbit. The second is a new algorithm necessary for guiding the execution of a skip trajectory for a lunar return scenario. Within the past year, two additional skip entry guidance algorithms have been provided to NASA by contractors and were implemented into ANTARES for performance comparison purposes.

A major focal point of guidance analysis for the Orion Project has been regarding skip guidance design. Skip guidance is considered to be at a low technology readiness level, but it is essential for meeting the requirements of the Orion Project. Key requirements are that the CEV must land on land within the continental United States and that it must be capable of returning from the moon at any time. The flight range needed to meet these requirements can only be met by flying a skip trajectory. The Apollo Project also derived (and flew) an algorithm to execute skip trajectories, but they had so little confidence in the algorithm that it was never executed on any mission. Since then, it has been determined that numerical predictor-corrector methods can perform the task with high probability of success. These approaches require significant computational power to execute, but the processor needs are now easily within the capacity of modern flight computers.

Still, due to the low technology readiness level of skip guidance algorithms, NASA elected to study the merits of the algorithms designed by both of the contractors that originally bid on the Orion Project. A three way comparison and merger effort was launched between NASA's Numerical Skip Entry Guidance (NSEG) algorithm, Lockheed Martin's Skip Guidance (LMSG) algorithm, and Draper Lab's PredGuid algorithm. All three were implemented into the same version of ANTARES so that sim-to-sim validation issues would not be a factor in the comparison. After the comparison was completed, the PredGuid algorithm was selected as the prime baseline algorithm for the Orion Project. The best features from the other two algorithms will be used to upgrade the prime algorithm as necessary to derive a final design. This approach proved to be highly successful and is a testament to the flexibility of the GN&C architecture that was designed into ANTARES from the start.

All of the skip guidance algorithms contain a version of the Apollo final phase guidance for the final atmospheric pass. Some are using versions close to the original Apollo design; others use a significantly modified and retuned version of the Apollo design. JSC created a modified Apollo guidance tuned for the CEV vehicle that is utilized as our primary option for low Earth orbit return trajectories. The final phase guidance will also need to be reconciled into a single CEV design in the future.

ANTARES also contains several simple guidance algorithms used for various applications. For example, attitude commands and rate commands are a form of guidance which must pass through the ANTARES guidance manager in order to get to the control manager. These simple guidance functions are used for the entry abort modes where a constant bank angle or a constant bank rate is commanded.

Finally, translational burn guidance and targeting algorithms are planned for the ANTARES entry simulation in the future to support features such as de-orbit burns, lunar trajectory trim burns, skip trajectory correction burns, and

the CM raise burn. Due to the uncertainty about the accuracy of the skip guidance algorithms, the Orion Project will need to design a capability to fire CM thrusters to correct the trajectory during the skip portion of a lunar return trajectory. Also, for low Earth orbit return trajectories, the CM trajectory must be raised after the Service Module is jettisoned in order to target the CM to the desired landing site while directing the SM to land in the ocean off the West Coast of the United States.

### **E. CEV Entry Control**

Three entry control algorithms have been implemented into ANTARES to date. The first was a perfect controller where the actual orientation state in dynamics is set perfectly equal to the commanded orientation. This reduces the simulation to a three degree-of-freedom translational dynamics simulator, but is useful for quick looks at guidance trajectory design. A later version of a perfect forced controller was attempted to model realistic rate and acceleration limits to better model large angle rotation maneuvers, but this model proved to be cumbersome and did not work very well. The second controller was a simple proportional-derivative controller coupled with a perfect effector to apply commanded torques directly to the vehicle. This controller provided the initial six degree-of-freedom simulation capability for the ANTARES entry simulation. It is still available to users today. This model also produces good control performance when coupled with a jet select logic that can transform commanded torques into equivalent RCS firings. This was done for the CEV Service Module, which is the current production controller for that vehicle in the ascent and on-orbit simulations. For the entry control problem, a third algorithm was implemented based on the design approach taken for the Apollo attitude controller. This algorithm performs bank angle control with damping of rates in the angle-of-attack and angle-of-sideslip axes.

The current CEV entry controller calculates attitude errors and body rate errors in the stability axis frame, which is defined with respect to the free-stream wind velocity vector. The stability axis frame is the angle-of-attack, angle-of-sideslip, bank angle (alpha-beta-bank) frame. It controls the bank angle based on bank angle commands provided by entry trajectory guidance. The alpha and beta angular rates are actively damped to protect against atmosphere and/or wind dispersions. Two options have been implemented for the jet select logic. The first is a bang-bang approach with a Schmitt Trigger. The second is a traditional phase-plane approach. An improvement to the controller was subsequently added to formalize the frame transformations in the controller in order to enable large angle maneuvers while outside of the atmosphere. This maneuvering capability is needed to support trajectory correction burns following separation from the Service Module and during the exo-atmospheric phase of a skip trajectory. This control algorithm has also been shown to work well as a bank rate controller for the emergency ballistic entry mode.

The CEV prime contractor has an independently designed entry controller that is now under design review but has not been implemented into ANTARES. The current NASA controller is the baseline design for performance analysis. Features from the prime contractor's design will be merged into the current controller in the future to produce one design to carry forward for the Orion Project.

## **V. Simulation Analysis for Orion Entry, Descent, & Landing**

The preceding sections covered in detail the overall design of the ANTARES entry simulation. The simulation architecture, the dynamics package, the GN&C architecture and design, and the numerous supporting component models were discussed. This section will discuss some unique aspects of how ANTARES is utilized for CEV entry performance analysis.

### **A. Monte Carlo Analysis**

Performing a controlled atmospheric entry of a space vehicle to a targeted landing site is a task with a great deal of variability and uncertainty. We will never have perfect knowledge of the atmosphere and aerodynamics or our ability to sense the environment. The design must be robust to a certain level of statistical certainty. Monte Carlo analysis techniques are used to prove the robustness of the design to expected dispersions within the CEV system and within the external environment. Large sets of runs are tested against specified random dispersion levels to create a sample size sufficient to define a statistical certainty of success.

The Trick simulation architecture utilized by ANTARES provides an excellent mechanism for automating the generation of Monte Carlo simulation samples. The Trick Monte Carlo capability can be configured through a single set of data file inputs such that one application can be launched to spawn a specified number of child processes to test a series of dispersions. The child processes can be distributed across multiple processors to efficiently complete large numbers of runs. The Monte Carlo data input file specifies the variables to be dispersed and provides instructions as to whether the dispersion will be done uniformly random, by Gaussian random

distribution, or by user-provided file data. For the random options, the user can specify the minimum/maximum values, the Gaussian sigma level, etc. The output from each run is stored in an individual storage directory created for each child process. Trick also provides post-processing tools for comparison plotting the results of the complete set of runs.

The Trick Monte Carlo capability produces a repeatable output. The exact same dispersion set can be re-run if needed. It also provides the ability to re-run a subset of runs or single runs within the set with the exact same dispersion set generated during the full execution sweep. We often run 3,000 case Monte Carlo sets to achieve the proper certainty of the performance results. We have learned to not record too much data because of limitations on hard disk storage capacity. We also found that the Trick post-processing tools are inadequate for processing very large sets of data. So, we have developed tools for transferring Trick output data to MatLab<sup>15</sup> for post-processing.

## B. Dispersion Sets

Table 1 summarizes the input data that is currently dispersed when we perform Monte Carlo analysis of CEV entry performance. It has been our philosophy to require the organizations that provide data for analysis to specify the appropriate dispersion approach and dispersion level. Through Trick, most any input data has the capability of being dispersed. The table is actually a subset of the most commonly dispersed parameters in the ANTARES entry simulation. Note that the table shows how we disperse the entry interface state. As discussed earlier, we also have the capability to disperse the EI state and navigation state via a covariance matrix. With this approach, the Monte Carlo input file would randomize a model seed for this initialization.

Parameter	Dispersion	Mean	3-sigma or Min/Max
EI Altitude	Uniform	400,000 ft	+/- 500 ft
EI Latitude/Longitude	Uniform	Varies	+/- 0.2 deg
EI Velocity	Gaussian	36,046 ft/sec (lunar) 25,859 ft/sec (LEO)	90 ft/sec
EI Flight Path Angle	Gaussian	-5.86 deg (lunar) -1.4 deg (LEO)	0.1 deg
EI Azimuth	Uniform	Varies	+/- 0.05 deg
Atmosphere and Winds	GRAM-99 model seed	--	--
Aero Coefficients	Defined by CAP team <sup>13</sup>	--	Varies through trajectory
Vehicle mass	Uniform	LM606 configuration <sup>14</sup>	+/- 3.1%
Vehicle center of mass	Gaussian	LM606 configuration	0.5 in – Xcg 0.3 in – Ycg, Zcg
Vehicle inertia	Gaussian	LM606 configuration	10% - Ixx, Iyy, Izz 50% - Ixy, Ixz, Iyz (correlated seeds)
RCS thrust	Gaussian	160 pounds	+/- 15%
GPS noise	Gaussian	--	Model seed
IMU noise	Gaussian	--	Model seed
IMU alignment position	Uniform	Varies	+/-0.1 in
IMU alignment attitude	Uniform	Varies	+/-20 arc sec

**Table 1. CEV monte carlo dispersion levels.**

## C. Support of Linear Stability Analysis

The ANTARES entry simulation is utilized for time domain analysis of CEV GN&C performance. The GN&C team utilizes separate tools for linear stability analysis within the frequency domain. Our linear analysis tool was developed within the MatLab application. It includes a linearized model of the CEV equations of motion at a selected flight condition with a linearized model of the CEV control system. The linear analysis tool relies on ANTARES to provide it flight condition data at discrete points along various entry trajectories.

Two post-processing capabilities have been implemented in ANTARES to support linear analysis. The first is a function that calculates the vehicle aerodynamic derivatives using a central differencing technique. The second is a

flight condition snapshot tool that enables the user to specify the trajectory conditions at which the data will be output to a file. The snapshot tool can trigger off of time, Mach number, altitude, dynamic pressure, or a user-defined external trigger. The aerodynamic derivatives can be calculated along with the flight condition snapshot. The output file is a text data file that can be loaded directly into the MatLab linear analysis tool.

#### **D. Verification and Validation**

The ANTARES entry simulation will serve as an independent assessment tool for the Orion Project. The CEV prime contractor will also develop and maintain an independent CEV GN&C simulation tool. The project plans to perform regular sim-to-sim validation to assure that both tools are equally valid for GN&C design and performance analysis. The goal is to have each simulation serve as a cross check of the other. During the CEV flight testing project, math models will be continuously improved based on measured data taken during the flight tests.

Throughout the ANTARES development process, a structured process of verification and validation is executed for every new capability that is implemented. Math models are verified against independent data whenever possible. Models that are received from outside sources are tested against simulation data provided by the outside source. Rigorous unit testing and integrated testing is performed. The integrated simulation also provides a set of regression test cases that enables all users to verify the simulation is executing as intended. The regression tests also identify the impact simulation changes have across all flight regimes (ascent, entry, orbit, lunar orbit, etc.). For major capabilities, a follow-on independent verification of the model is performed by a person other than the original developer.

A large part of the ANTARES simulation software has been leveraged from other NASA Trick simulations for programs such as Space Shuttle, Space Station, etc. These models have a long history of successful use and the user community has great confidence in their pedigree. Extensive use of delivered software has been made to utilize the expertise of domain experts. We rely heavily on the domain experts to certify the performance of their models in ANTARES prior to release of the model to the user community.

### **VI. Conclusion**

The ANTARES entry simulation has been successfully created over the past two years to serve the time domain six degree-of-freedom simulation needs of the Orion Entry, Descent, and Landing team. The primary purpose of the simulation tool is to support GN&C design and analysis. However, it is expected to serve as the primary simulation tool for many applications at various NASA centers ranging from engineering analysis, mission operations, crew training, avionics-in-the-loop testing, and so on.

ANTARES has taken advantage of many advances in simulation technology that have been in development supporting other NASA space vehicle programs such as Space Shuttle and Space Station. Substantial amounts of simulation software and GN&C algorithms were successfully adapted or reused from other NASA simulation tools. This allowed the very quick development of a very capable CEV GN&C simulation tool. It has already been successfully used for several rounds of GN&C requirements analysis, system definition analysis, and early design analysis for the Orion Project. It has demonstrated flexibility in simulation architecture which has enabled continuous evolution in capability and fidelity to meet the needs of the CEV GN&C design and analysis community.

#### **Appendix – Example Simulation Output**

The ANTARES output results for one example simulation run are presented in this appendix. The run represents the lunar return entry trajectory for a long skip scenario where the CEV vehicle must travel 5250 nautical miles downrange and 250 nautical miles cross range from the Entry Interface point to the landing target at the Utah Test Range. The CEV CM is the LM606 configuration for mass properties and RCS configuration. The environment is the mean GRAM-99 atmosphere and the CEV Version 0.25 aerodynamics API. Navigation sensors are the IMU and GPS with GPS signals reacquiring nominally after each blackout period. The GN&C is configured to the high-fidelity navigation algorithm, the NASA Numerical Skip Entry Guidance algorithm, and the entry bang-bang controller. The initial EI state is as follows:

Geodetic Altitude = 400,000 ft  
Geodetic Latitude = -46.5 deg  
Longitude = -112.552 deg  
Inertial Velocity = 36,046 ft/sec  
Inertial Flight Path Angle = -6.0 deg  
Inertial Azimuth Angle = 0.0 deg

Angle of Attack = 159.0 deg  
Bank angle = 65.0 deg  
Angle of Sideslip = 0.0 deg  
Angular Body Rate = 0.0 deg/sec

Figure 1 show the translational trajectory of for the run. Figure 2 shows the attitude history. Figure 3 shows the control performance. Figure 4 shows the guidance performance. Figure 5 shows the aerothermal performance.



CEV translation  
(SIM\_entry/RUN\_cev\_entry\_lunar\_skip)

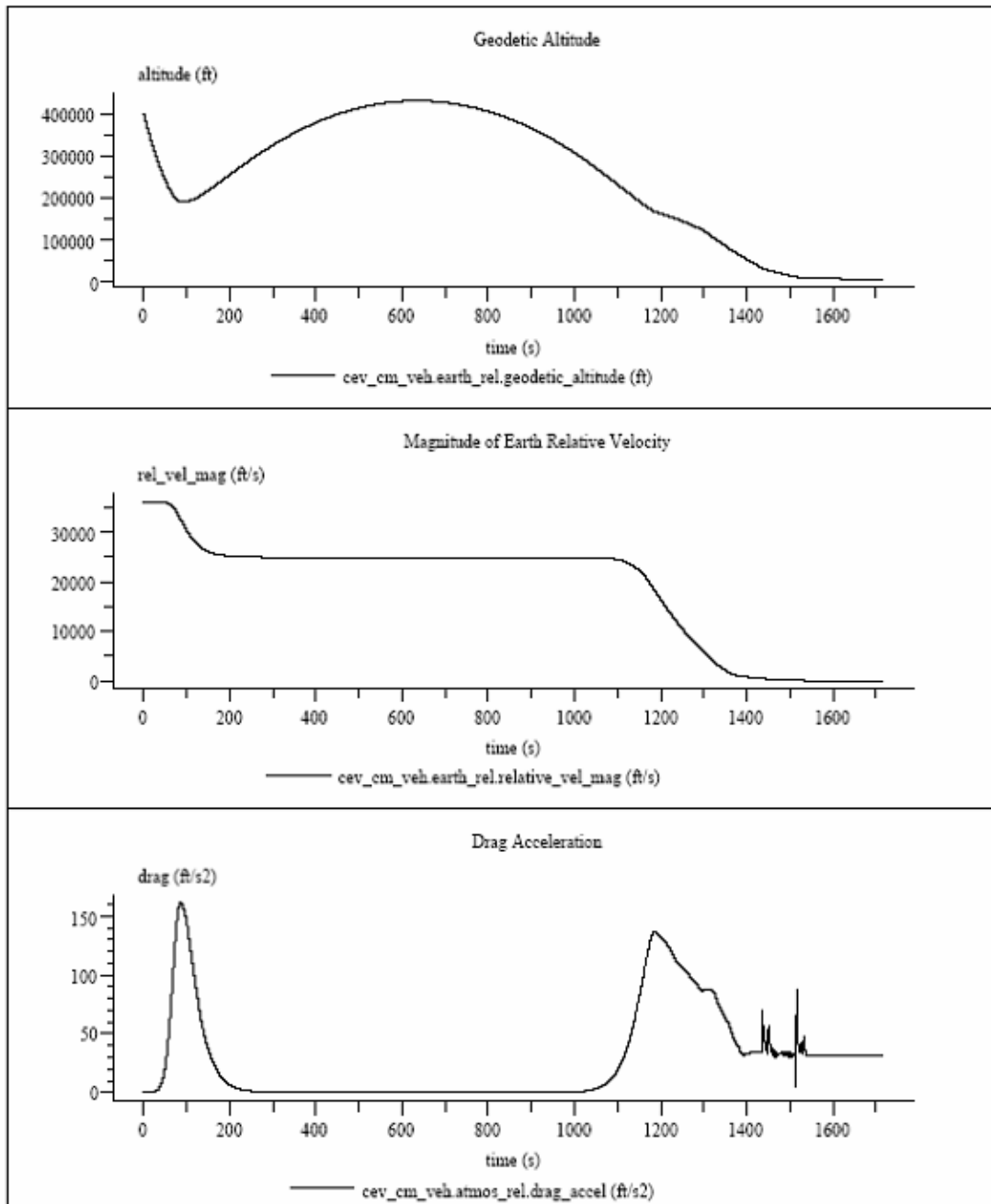
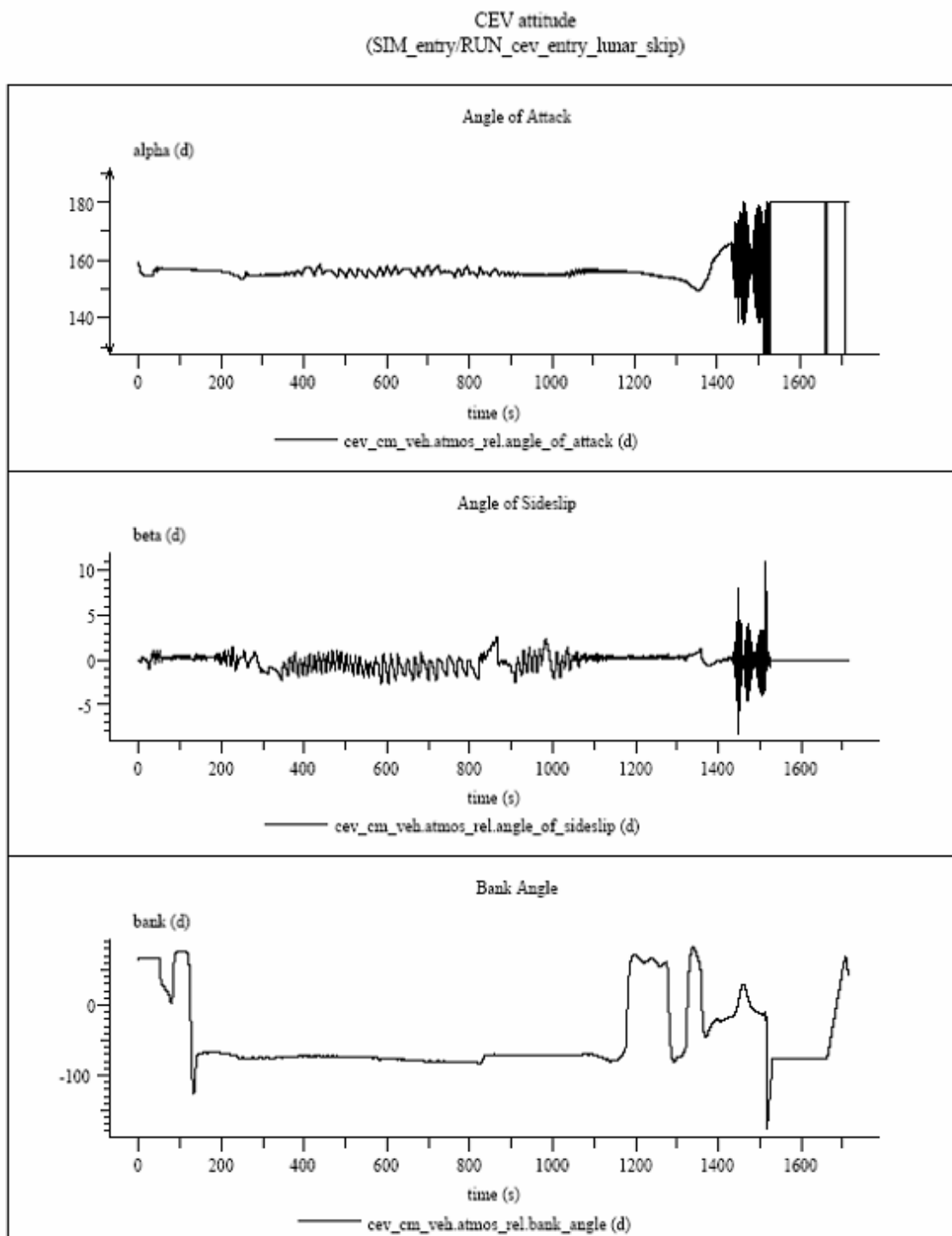
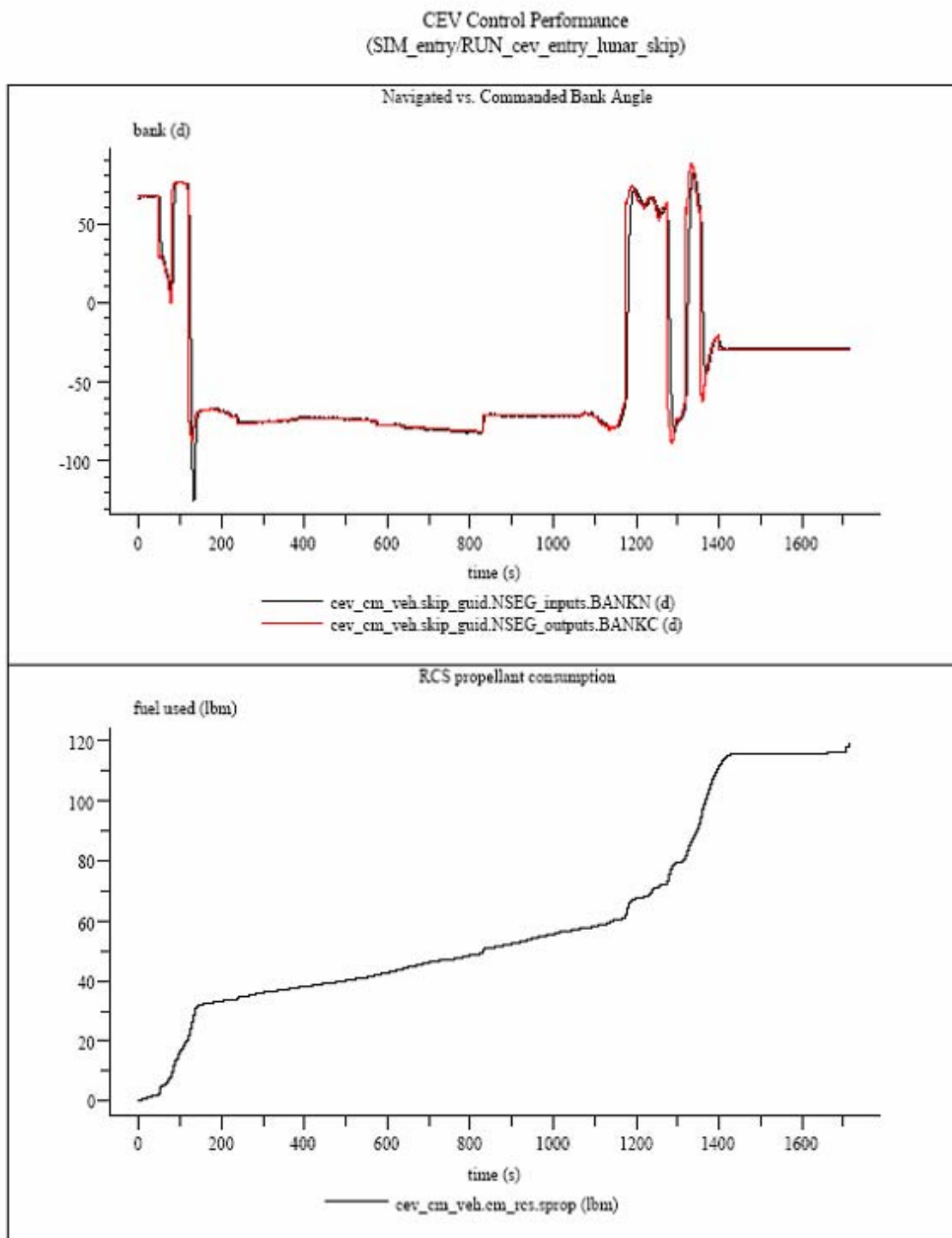


Figure 1. CEV trajectory time history.



**Figure 2. CEV attitude time history.**



**Figure 3. CEV control performance.**

CEV Trajectory Performance  
(SIM\_entry/RUN\_cev\_entry\_lunar\_skip)

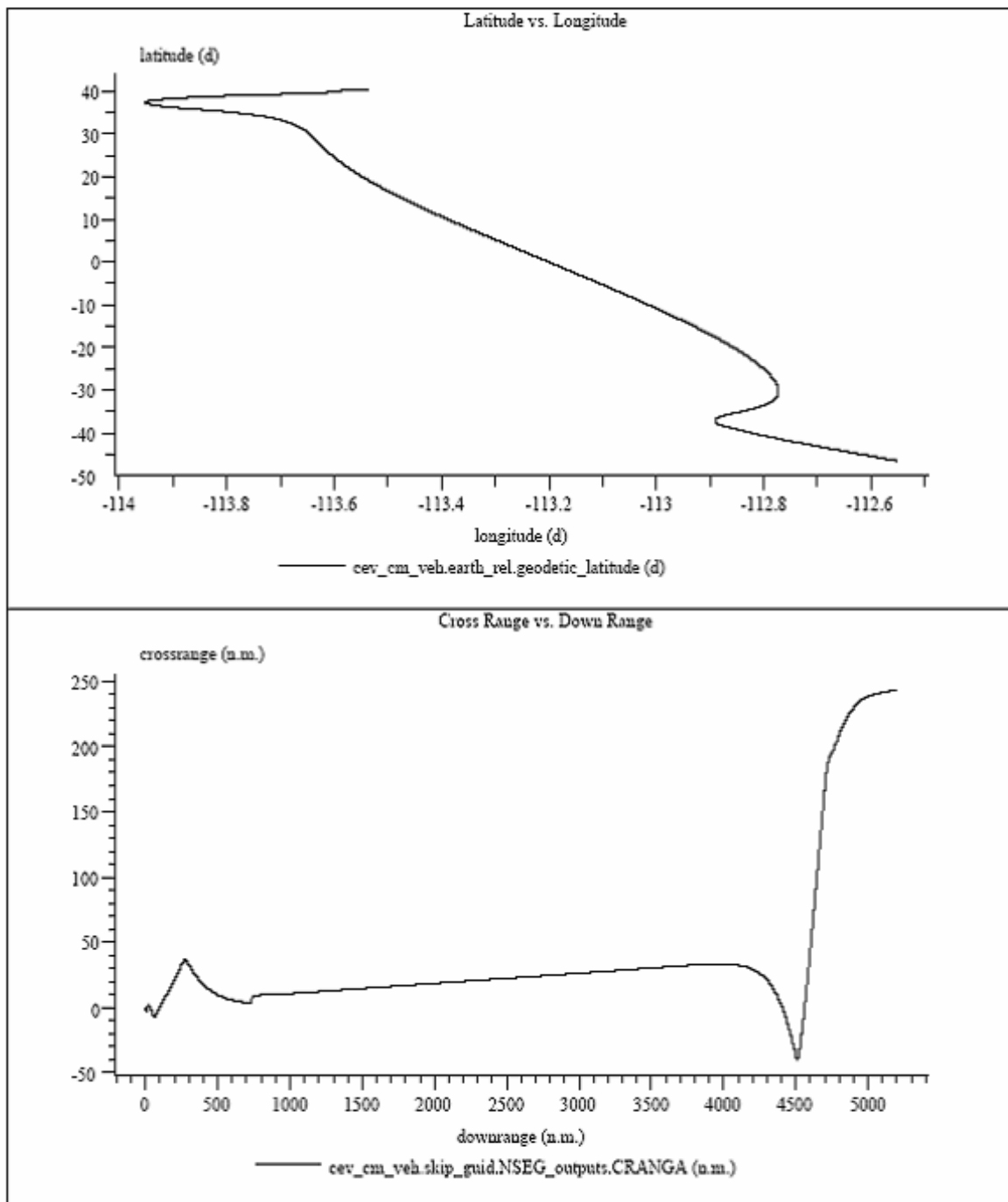


Figure 4. CEV guidance performance.

CEV Aerothermal Performance  
(SIM\_entry/RUN\_cev\_entry\_lunar\_skip)

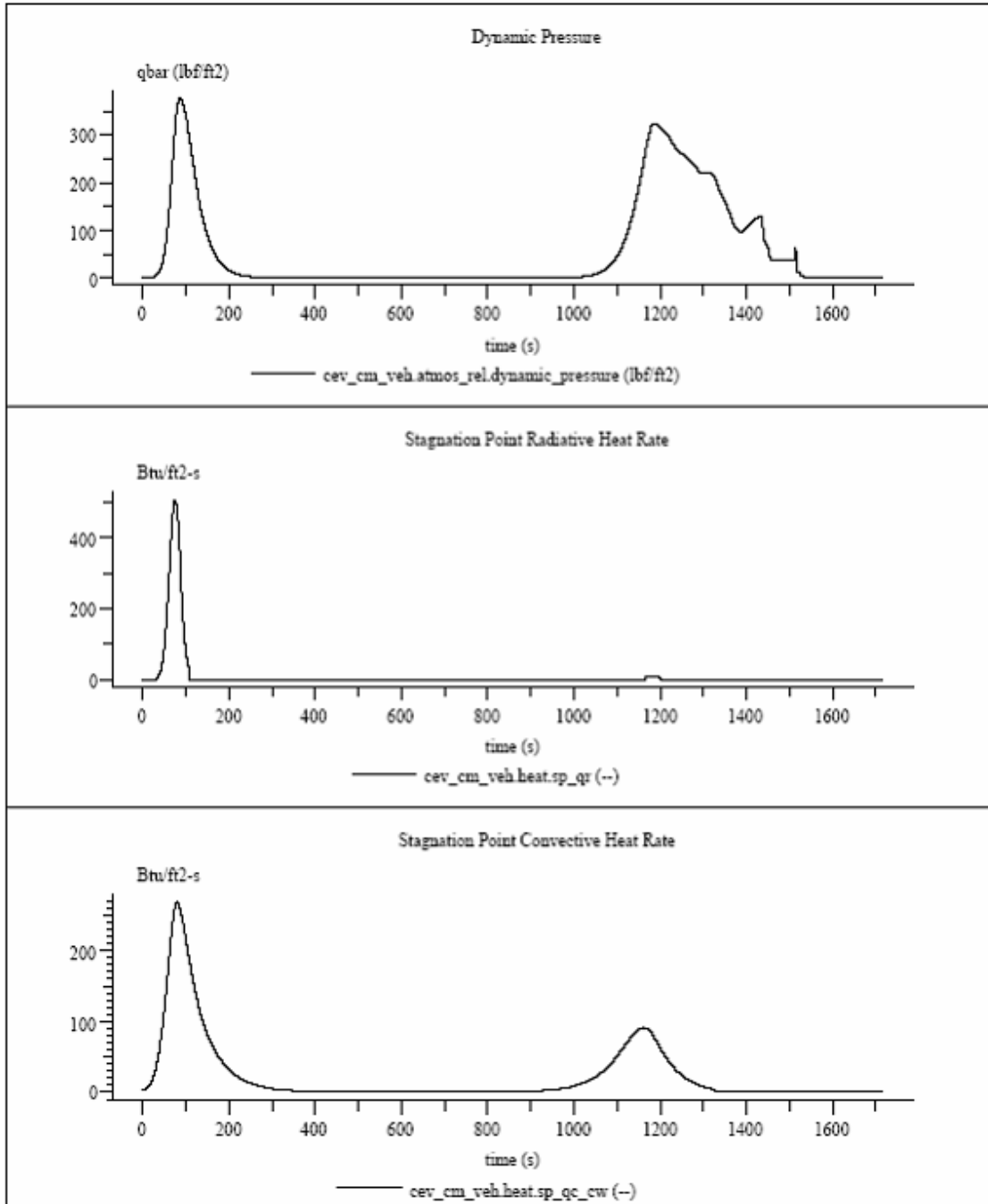


Figure 5. CEV aerothermal performance.

## Acknowledgments

Many people have played a role in the creation of the ANTARES entry simulation. The author wishes to acknowledge the particular significant contributions made by the following colleagues - Brian Bihari, Jeremy Rea, Robert Gay, Edgar Medina, Susan Stachowiak, Amanda Acevedo, and Jason Arnold.

## References

- <sup>1</sup>ClearCase, Version 2003.06.00, Rational Software, IBM Corp., Armonk, New York, 2003.
- <sup>2</sup>ClearDDTS, Version 4.8, Rational Software, IBM Corp., Armonk, New York, 1997, 2003.
- <sup>3</sup>Trick Generic Simulation Toolkit, JSC Automation, Robotics, and Simulation Division, Houston, Texas, 1997-2003.
- <sup>4</sup>JEOD, JSC Engineering Orbital Dynamics Package, JSC Automation, Robotics, and Simulation Division, Houston, Texas, 2007.
- <sup>5</sup>EDGE, Engineering DOUG Graphics for Exploration Package, JSC Automation, Robotics, and Simulation Division, Houston, Texas, 2007.
- <sup>6</sup>The NASA GSFC & NIMA Joint Geopotential Model, Version EGM96, Lemoine, F.G., et.al., NASA Goddard Space Flight Center, Greenbelt Maryland, July 1998.
- <sup>7</sup>Earth RNP for the Standard Epoch J2000, JSC-24574, Mulcihy, D.D., and Bond, V.R., NASA Johnson Space Center, Houston, Texas, September, 1990.
- <sup>8</sup>JPL Planetary and Lunar Ephemerides, Version DE405, Standish, E.M., et.al., NASA Jet Propulsion Laboratory, Pasadena, California, June 1997.
- <sup>9</sup>Orion Aerodynamics API, Version 0.25, Greathouse, J., and Thompson, J.A., NASA Johnson Space Center, Houston, Texas, May 2007.
- <sup>10</sup>GRAM-99, Global Reference Atmosphere Model, Version 3, Justus, C.G., and Leslie, F.W., NASA Marshall Spaceflight Center, Huntsville, Alabama, October 2006.
- <sup>11</sup>CEV Entry Heating Model, Version 3.2, Wurster, K., NASA Langley Research Center, Hampton, Virginia, June 2007.
- <sup>12</sup>GN&C Architecture Design for ARES Simulation, Revision 3.0, Gay, R.S., NASA Johnson Space Center, Houston, Texas, June 2006.
- <sup>13</sup>Orion Aerodynamic Databook, Version 0.25, Draft 2, CXP-72167, NASA Johnson Space Center, April 2007.
- <sup>14</sup>Orion Vehicle Simulation Data Book, Preliminary Issue, Revision A, CEV-MA-07-003, Lockheed Martin Space Systems Company, Denver, Colorado, April 2007.
- <sup>15</sup>MATLAB, Version 7.4, The MathWorks, Inc., Natick, Massachusetts, 1994-2007.